

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-223139

(43)Date of publication of application : 12.08.1994

(51)Int.Cl.

G06F 15/60

(21)Application number : 05-029773

(71)Applicant : NEC CORP

(22)Date of filing : 26.01.1993

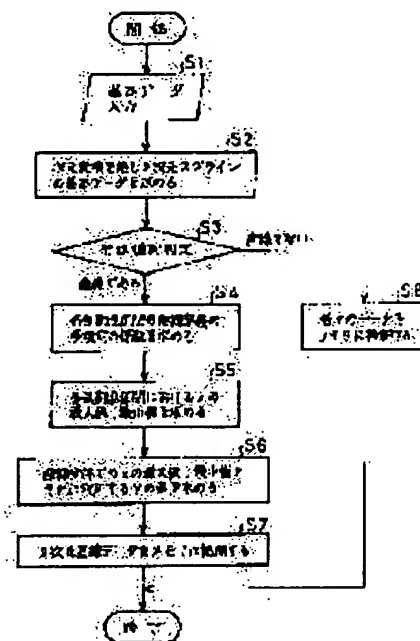
(72)Inventor : TAKAHASHI HIROYUKI

(54) METHOD FOR DECIDING LINEAR SHAPE IN DIMENSION CONVERSION OF THREE-DIMENSIONAL SPLINE

(57)Abstract:

PURPOSE: To specify the start point and end point of a straight line even unless control points are in a line if two-dimensional B-spline curve data obtained from three-dimensional B-spline curve data through dimension conversion indicate a straight line.

CONSTITUTION: In S1, control point information and knot vectors being the basic data of the three-dimensional B-spline curve data are inputted and a projecting method, a conversion matrix, etc., are inputted. In S2, the conversion matrix is used to convert the three-dimensional spline curve data into the two-dimensional B-spline curve data. When it is decided that the shape is a straight line in S3, the coefficients of a curve definition polynomial by control point sections are found in S5, maximum and minimum values of (x) in the respective control point sections are found in S5, and values (y) corresponding to the maximum and minimum values of (x) in all the control point sections are found in S6. Then the maximum and minimum values of (x) in all the control point sections and the values (y) corresponding to them are stored in a memory as straight line data showing the start and end points of the straight line in S7.



LEGAL STATUS

[Date of request for examination]

26.01.1993

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

2570954

[Date of registration]

24.10.1996

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-223139

(43)公開日 平成6年(1994)8月12日

(51)Int.Cl.⁵

G 0 6 F 15/60

機別記号

4 0 0 A 7623-5L

庁内整理番号

F I

技術表示箇所

審査請求 有 請求項の数 2 F D (全 14 頁)

(21)出願番号 特願平5-28773

(22)出願日 平成5年(1993)1月26日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 高橋 裕幸

東京都港区芝五丁目7番1号 日本電気株式会社内

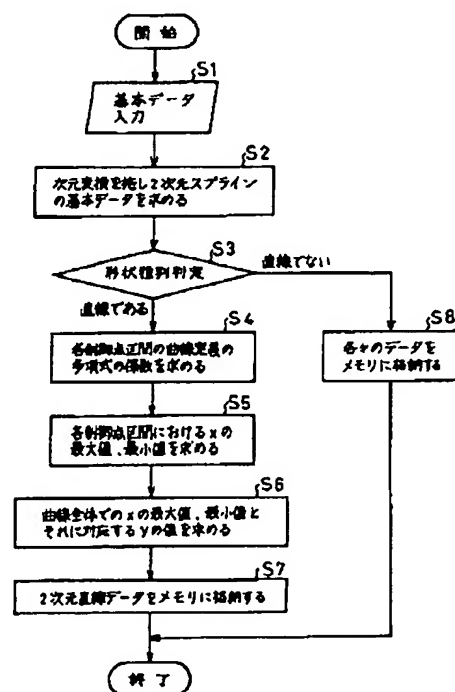
(74)代理人 弁理士 境 廣巳

(54)【発明の名称】 3次元スプラインの次元変換時の直線形状判定方法

(57)【要約】

【目的】 3次元B-スプライン曲線データを次元変換した2次元B-スプライン曲線データが直線を示すものになった場合、制御点が直線上に並んでいなくても、直線の始点、終点を特定できるようにする。

【構成】 S1で3次元B-スプライン曲線データの基本データである制御点情報、ノットベクトルを入力すると共に、投影方法や変換マトリックス等を入力する。S2で、変換マトリックスを用いて、3次元B-スプライン曲線データを2次元B-スプライン曲線データに変換する。S3で形状が直線であると判定された場合は、S4で各制御点区間毎の曲線定義多項式の係数を求め、S5で各制御点区間に於けるxの最大値、最小値を求め、S6で全制御点区間に於けるxの最大値、最小値とそれに対応するyの値を求め、S7で全制御点区間に於けるxの最大、最小値とそれに対応するyの値とを直線の始点、終点を示す直線データとしてメモリに格納する。



【特許請求の範囲】

【請求項1】 3次元B-スプライン曲線データに対して次元変換を行なうことにより、次元変換後の2次元B-スプライン曲線データが直線を示すものとなり、且つ制御点が直線上に順列に並んでいないものになった場合、

各制御点区間に曲線定義多項式とその1次導関数の式とを導き、それらを用いて各制御点区間毎の最大値、最小値を算出し、
前記各制御点区間毎の最大値、最小値に基づいて全制御点区間に於ける最大値、最小値を算出し、
前記全制御点区間に於ける最大値、最小値に基づいて前記直線の始点、終点を特定することを特徴とする3次元スプラインの次元変換時の直線形状判定方法。

【請求項2】 前記各制御点区間毎の最大値、最小値は、各制御点区間の端点の値と各制御点区間の極大値、極小値とに基づいて求めることを特徴とする請求項1記載の3次元スプラインの次元変換時の直線形状判定方

法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は3次元CADシステム等の3次元システム上の3次元B-スプライン曲線データを2次元CADシステム等に移行させるための次元変換時に行なう要素種別判定方法に関し、特に次元変換後の直線形状を判定する3次元スプラインの次元変換時の直線形状判定方法に関する。

【0002】

【従来の技術】 B-スプライン曲線は2次元、3次元を問わず、図10に示すように、4つの制御点 P_0, P_1, P_2, P_3 と8つのノットベクトル $(t_0, t_0, t_0, t_0, t_1, t_1, t_1, t_1)$ が最小の構成単位となっており、この区間の曲線定義式は、ノットベクトルによって算出されるスプライン関数 $N_0(t), N_1(t), N_2(t), N_3(t)$ を用いて次式(1)に示すように表される。

【0003】

$$P_0 N_0(t) + P_1 N_1(t) + P_2 N_2(t) + P_3 N_3(t) \dots (1)$$

但し、 $t \in [t_0, t_1]$

【0004】ところで、B-スプライン曲線データが示す線分が直線である場合、そのことを判定し、直線の始点、終点の座標値を求めることが必要になる場合がある。従来、B-スプライン曲線データによって示される線分が直線であるか否かの判定は、隣接するベクトル $(P_{i+1} - P_i), (P_{i+2} - P_{i+1})$ どうしの外積を計算し、各軸方向の成分が0であるか否かに基づいて直線であるか否かを判定している(例えば、特開昭63-211071号公報)。また、この時、直線の基本データである始点、終点の座標値は制御点の始点、終点の座標値となる。

【0005】また、3次元CADシステム上の3次元B-スプライン曲線データを2次元CADシステムに移行するために次元変換を行なった後、次元変換後の2次元B-スプライン曲線データによって示される線分が直線か否かを判定し、直線の始点、終点の座標値を求める方法としては、図7に示すような方法が考えられる。以下、図7に示す方法を詳細に説明する。

【0006】オペレータによって3次元CADシステム上の3次元B-スプライン曲線が指示され、且つ投影方法が入力されると、S61でデータベース内に格納されている上記3次元B-スプライン曲線の基本データ(制御点情報とノットベクトル)及び投影面への変換マトリックスを読み込む。

【0007】次にS62で、投影方法が平行投影か透視投影かを判定する。平行投影と判定した場合は、S63で、各制御点の座標値を変換マトリックスで変換して2次元座標の部分である (x, y) だけを取り出し、これと最初に入力したノットベクトルとを新2次元B-スプライン曲線の基本データとする。また、透視投影と判定

した場合は、S64で基本データに基づいて通過点の座標値を求め、S65で通過点の座標値を変換マトリックスで変換して2次元座標の部分である (x, y) だけ取り出し、新通過点とする。S66でこの新通過点により新たに2次元B-スプライン曲線を張り直し、その基本データを得る。

【0008】次に新2次元B-スプライン曲線の基本データである制御点座標値により、その形状判定を行なう。先ず、S67で2次元B-スプライン曲線が1点になるかどうかを図8の流れ図に沿って判定する。

【0009】図8では先ず、S81で2次元B-スプライン曲線の基本データ(制御点情報とノットベクトル)を受け取り、次のS82で、各制御点 $P_0, P_1, \dots, P_i, \dots$ を頂点とする多角形の重心を求める。

【0010】その後、S83で、変数 i に0をセットし、変数 i の値が(制御点数-1)以上になるか、或いはS89で2次元点でないかと判定するまで、以下の処理を行なう。

【0011】先ず、S85で制御点 P_i と重心との間の距離を求め、S86でその距離が誤差範囲かどうかをチェックする。一定の誤差内に収まっている場合はS87で変数 i に1を加えて処理を継続し、そうでない場合はS89で、2次元点ではないという情報をメインの処理に返し、2次元B-スプライン曲線が1点になるかどうかを判定する処理を終える。また、全ての制御点と重心との間の距離が誤差範囲内である場合(S84がNOの場合)は、2次元点データ(重心の座標値)をメインの処理に返し(S88)、処理を終了する。

【0012】図7のメインの処理では2次元点データが返されることにより、S68で2次元点データをメモリ(図示せず)に格納し、処理を終了する。また、2次元

点ではないという情報が返された場合は、S69で、直線になるかどうかの判定を行なう。

【0013】直線になるかどうかの判定は図9に示す流れ図に沿って、以下に行なう。

【0014】S91で、2次元B-スプライン曲線の基本データ（制御点情報とノットベクトル）の受け取りを行ない、S92で、変数 i に0をセットする。

【0015】その後、S93で、各制御点に対する方向ベクトル $P_0, P_1, \dots, P_i, \dots$ に対し、ベクトル V_i とその大きさ L_i ($i=0, 1, 2, \dots$, 制御点数-1) とをそれぞれ次式(2)、(3)のように定義する。

$$【0016】V_i = P_{i+1} - P_i \quad \dots (2)$$

$$L_i = \|V_i\| \quad \dots (3)$$

【0017】以下、 i の値が（制御点数-1）以上になるか（S94がNO）、或いはS103で2次元直線ではないと判定するまで、以下の処理を行なう。

【0018】S95で、ベクトル V_{i+1} とその大きさ L_{i+1} とをそれぞれ次式(4)、(5)に示すように定義する。

$$【0019】V_{i+1} = P_{i+2} - P_{i+1} \quad \dots (4)$$

$$L_{i+1} = \|V_{i+1}\| \quad \dots (5)$$

【0020】S96で、 L_i と L_{i+1} を一定の誤差 ε_1 と比較して(6)、(7)、(8)のように分類する。

$$【0021】L_i < \varepsilon_1 \rightarrow \text{Case 4} \quad \dots (6)$$

$$L_i > \varepsilon_1 \cap L_{i+1} < \varepsilon_1 \rightarrow \text{Case 5} \quad \dots (7)$$

$$L_i > \varepsilon_1 \cap L_{i+1} > \varepsilon_1 \rightarrow \text{Case 6} \quad \dots (8)$$

【0022】Case 4の場合は、S97で $V_i \leftarrow V_{i+1}$ 、 $L_i \leftarrow L_{i+1}$ とし、S98で変数 i に1を加え、変数 i の値が（制御点数-1）未満の場合（S94がYESの場合）はS95で新たな V_{i+1} 、 L_{i+1} を定義し、S96で前述したと同様の分類処理を行なう。

【0023】Case 5の場合は、 V_i 、 L_i はそのまま、S98で変数 i に1を加え、変数 i の値が（制御点数-1）未満の場合（S94がYESの場合）はS95で新たな V_{i+1} 、 L_{i+1} を定義し、S96で前述したと同様の分類処理を行なう。

【0024】Case 6の場合は、まず、S99でベクトル V_i 、 V_{i+1} を次式(9)、(10)のように正規化して単位ベクトル n_i 、 n_{i+1} を求める。

$$【0025】n_i = V_i / L_i \quad \dots (9)$$

$$n_{i+1} = V_{i+1} / L_{i+1} \quad \dots (10)$$

【0026】次に、S100で内積 $N = n_i \cdot n_{i+1}$ と、外積 $G = n_i \times n_{i+1}$ を計算し、S101で次式(11)を満足させるか否かを判断する。

$$【0027】N = 1 \cap G = 0 \quad \dots (11)$$

【0028】尚、式(11)に於いて内積 $N = 1$ は制御点 P_i 、 P_{i+1} 、 P_{i+2} が直線上に順列に並んでいるか否かを判定するための条件であり、外積 $G = 0$ は制御点 P_i 、 P_{i+1} 、 P_{i+2} が直線上にあるか判定するための条件である。

【0029】そして、 $N = 1 \cap G = 0$ の時、即ち、制御点 P_i 、 P_{i+1} 、 P_{i+2} が直線上に順列に並んでいると判定した場合は、S102で $V_i \leftarrow V_{i+1}$ 、 $L_i \leftarrow L_{i+1}$ とし、S98で変数 i に1を加え、変数 i の値が（制御点数-1）未満の場合（S94がYES）はS95で新たに V_{i+1} 、 L_{i+1} を定義し、S96で前述したと同様の分類処理を行なう。

【0030】また、 $N = 1 \cap G = 0$ でない場合は、S103で2次元直線ではないと判定し、その情報をメインの処理に返し、直線判定の処理を終了する。

【0031】また、S94で変数 i の値が（制御点数-1）以上になったと判断した場合は、S104でメインの処理に制御点の始点、終点の座標値をそれぞれ直線の始点、終点の座標値として返し、処理を終了する。

【0032】図7のメインの処理では、2次元直線データ（直線の始点、終点の座標値）が返されると、S70でそれをメモリに格納する。また、2次元直線ではないという判定結果が返された場合は、S71で、S63或いはS66で作成した2次元B-スプライン曲線データをメモリに格納する。

【0033】

【発明が解決しようとする課題】図7に示した方法で直線と判定するのは、図11に示すように制御点が1直線上に順列に並んでいる場合であり、前述したように、式(11)の内積 $N = 1$ がこの順列並びを保証する条件になっている。実はこれ以外にも直線となる場合として図12に示すように制御点が直線上に順列に並んでいない場合が考えられる。この場合も含めて直線と判定するためには式(11)の内積 $N = 1$ に関する部分を取り去り、単に $G = 0$ とすれば良いが、図12に示すように制御点が直線上に順列に並んでいない場合は直線の基本データである始点、終点の座標値を求めることができないため、やむなく2次元B-スプライン曲線と判定し、2次元B-スプライン曲線の基本データを返すようにしなければならなかった。

【0034】また、特開昭63-211071号公報の方法は制御点の始点、終点の座標値を直線の始点、終点の座標値としているため、この方法によって、3次元B-スプライン曲線データを次元変換して得られた2次元B-スプライン曲線データが示す線分が直線か否かを判定し、直線の始点、終点の座標値を求めるとすると、図12に示すように制御点が直線上に順列に並んでいない場合、始点、終点の座標値が誤ったものになってしまうという問題がある。

【0035】本発明の目的は、制御点が順列に並んでいない2次元直線についても、2次元直線の基本データである始点、終点の座標値を求め、正確な直線データを定めることができる3次元スプラインの次元変換時の直線形状判定方法を提供することにある。

【0036】

【課題を解決するための手段】本発明は上記目的を達成するため、3次元B-スプライン曲線データに対して次元変換を行なうことにより、次元変換後の2次元B-スプライン曲線データが直線を示すものとなり、且つ制御点が直線上に順列に並んでいないものになった場合、各制御点区間毎に曲線定義多項式とその1次導関数の式とを導き、それらを用いて各制御点区間毎の最大値、最小値を算出し、前記各制御点区間毎の最大値、最小値に基づいて全制御点区間に於ける最大値、最小値を算出し、前記全制御点区間に於ける最大値、最小値に基づいて前記直線の始点、終点を特定するようにしたものである。

【0037】

【実施例】次に本発明の実施例について図面を参照して詳細に説明する。

【0038】図1は本発明の実施例のハードウェア構成を示す外観図である。本発明を実施するためのエンジニアリングワークステーション(EWS)10は、3次元B-スプライン曲線の次元変換時の形状判定の演算や、各デバイス装置に演算結果を知らせる役割を果たすCPUやメモリ等を内蔵した演算装置1と、CAD図面として曲線等の要素を表示出力するCRTディスプレイ2と、数字や文字をオペレータが入力するのに用いるキーボード3と、画面上の曲線の要素をオペレータが指定するのに用いるタブレット4及びスタイラスペン5とを備えている。

【0039】次に、演算装置10を用いて本発明を実行する手順について説明する。

【0040】本発明の3次元B-スプライン曲線の次元変換時の直線形状判定方法の要点は、図2の流れ図に示すように、オペレータがスタイラスペン5を用いてCRTディスプレイ2の画面上の3次元B-スプライン曲線を指示し、キーボード3を用いて投影方法を入力すると、S1で、3次元B-スプライン曲線の基本データ(制御点情報、ノットベクトル)と、変換に必要な情報(投影方法、変換マトリックス)とをデータベースから入力し、S2で、変換マトリックスに従って2次元B-スプライン曲線に次元変換してその基本データ(制御点情報、ノットベクトル)を求める。

【0041】更に、S3で2次元B-スプライン曲線の形状の種別を判定し、2次元点または2次元B-スプライン曲線になる場合はS8でそのデータを演算装置1内のメモリに格納し、処理を終了する。2次元直線になる場合はS4に進む。

【0042】S4では各制御点の区間の曲線を定義する多項式の係数を求める。次のS5では各制御点の区間に於けるx(またはy)の最大値、最小値を求める。更に、S6では各制御点の区間に於けるx(またはy)の最大値、最小値をもとに曲線全体でのx(またはy)の最大値、最小値とそれに対応するy(またはx)の値を算出する。最後のS7ではxの最小値とそれに対応する

yの値を始点座標値とし、xの最大値とそれに対応するyの値を終点座標値とした2次元直線データを演算装置1内のメモリに格納し、処理を終了する。

【0043】図3～図6は本発明の実施例の詳細な流れ図である。

【0044】オペレータがCRTディスプレイ2の画面上に表示されている3次元B-スプライン曲線をスタイラスペン5を使用して指示し、キーボード3によりその他の必要な情報を入力すると、演算装置1は図3のS11で、指示された3次元B-スプライン曲線の制御点座標やノットベクトル等の基本データと変換マトリックス等の変換に必要なデータとをデータベースから入力し、予め確保しておいた変数領域に格納する。

【0045】次に、S12で、前述した図7のS62～S66と同じ手順により3次元B-スプライン曲線データの次元変換を行ない、2次元B-スプライン曲線の基本データである制御点座標やノットベクトルのデータを求め、格納する。

【0046】その後、S13で前述した図8のS81～S89と同じ手順により2次元点になるかどうかの判定を行ない、2次元点になる場合はS14で2次元点データを演算装置1内のメモリに格納した後、処理を終了する。

【0047】更に、S15で直線判定フラグをOFFにし、S16で変数iに0をセットし、次のS17で各制御点 $P_0, P_1, \dots, P_i, \dots$ に対してベクトル V_i とその大きさ L_i とを式(12)、(13)に示すように定義する。

$$V_i = P_{i+1} - P_i \quad \dots (12)$$

$$L_i = \|V_i\| \quad \dots (13)$$

【0049】以下、変数iの値が(制御点数-1)を超えるまで(図4のS18がNOとなるまで)、或いは図4のS25で2次元直線ではない(S25でNO)と判定されるまで、以下の処理を行なう。

【0050】図4のS19で、各制御点に対する方向ベクトル $P_0, P_1, \dots, P_i, \dots$ に対し、ベクトル V_{i+1} とその大きさ L_{i+1} ($i=0, 1, 2, \dots, \text{制御点数}-1$)を式(14)、(15)に示すように定義する。

$$V_{i+1} = P_{i+2} - P_{i+1} \quad \dots (14)$$

$$L_{i+1} = \|V_{i+1}\| \quad \dots (15)$$

【0052】S20で L_i と L_{i+1} を一定の誤差 ε_1 と比較して次式(16)、(17)、(18)のように分類する。

$$L_i < \varepsilon_1 \rightarrow \text{Case 1} \quad \dots (16)$$

$$L_i > \varepsilon_1 \cap L_{i+1} < \varepsilon_1 \rightarrow \text{Case 2} \quad \dots (17)$$

$$L_i > \varepsilon_1 \cap L_{i+1} > \varepsilon_1 \rightarrow \text{Case 3} \quad \dots (18)$$

【0054】Case 1の場合は、S21で、 $V_i \leftarrow V_{i+1}$ 、 $L_i \leftarrow L_{i+1}$ とし、S22で変数iに1を加え、変数iの値が(制御点数-1)未満の場合(S18がYESの場合)はS19で新たな V_{i+1} 、 L_{i+1} を定義

し、S20で前述したと同様の比較分類を行なう。

【0055】Case2の場合は、 V_i, L_i はそのまま、S22で変数 i に1を加え、変数 i の値が(制御点数-1)未満の場合(S18がYESの場合)はS19で新たな V_{i+1}, L_{i+1} を定義し、S20で前述したと同様の比較分類を行なう。

【0056】Case3の場合は、まず、S23でベクトル V_i, V_{i+1} を次式(19)、(20)を用いて正規化し、単位ベクトル n_i, n_{i+1} を求める。

$$【0057】 n_i = V_i / L_i \quad \dots (19)$$

$$n_{i+1} = V_{i+1} / L_{i+1} \quad \dots (20)$$

【0058】次に、S24で内積 $N = n_i \cdot n_{i+1}$ と外積 $G = n_i \times n_{i+1}$ とを計算し、S25で外積 $G=0$ であるか否かを、即ち2次元直線となるか否かを判定する。

【0059】そして、外積 $G=0$ でないと判定した場合は、2次元スプライン曲線データとして基本データ(制御点情報、ノットベクトル)をメモリに格納して処理を終了する。

【0060】また、外積 $G=0$ であると判定した場合、即ち2次元直線になると判定した場合はS27で $V_i \leftarrow V_{i+1}, L_i \leftarrow L_{i+1}$ として新たな V_{i+1}, L_{i+1} を定義する。

【0061】その後、S28で内積 $N=1$ かどうかの判定を行ない、内積 $N=1$ の場合はS22で変数 i に1を加え、変数 i の値が(制御点数-1)未満の場合(S18がYES)は、S19で新たな V_{i+1}, L_{i+1} を定義し、S20で前述したと同様の比較分類を行なう。また、内積 $N=1$ でないと判定した場合はS29で直線判

定フラグにオンをセットした後、S22の処理を行なう。

【0062】以上の処理を繰り返し行ない、変数 i の値が(制御点数-1)以上となると、図5に示す処理が行なわれる。既にこの段階で、次元交換された結果は直線形状であると区別がついている。

【0063】図5のS30で、直線判定フラグの評価を行ない、それがOFFの場合はS31で制御点の始点、終点の座標値を直線の始点、終点の座標値として演算装置1内のメモリに格納し、データを返し、処理を終了する。

【0064】また、直線判定フラグがONの場合は、以下に示す処理を施し、直線データ(始点、終点の座標値)を算出する。

【0065】まず、S32で制御点の始点と終点とからこの直線形状が y 軸に平行であるかどうかを調べる。 y 軸に平行でない場合はS34で変数 i に0をセットし、平行な場合はS33で y 軸平行フラグをONにした後、S34で変数 i に0をセットする。

【0066】その後、変数 i が(制御点数-3)以上になるまで(S35がYESとなるまで)、以下の処理を行なう。

【0067】まず、図6のS43で制御点区間 i に於けるスプライン関数 $N^i_0(t), N^i_1(t), N^i_2(t), N^i_3(t)$ を求める。次のS44でスプライン関数と制御点座標値 $P^i_x, P^{i+1}_x, P^{i+2}_x, P^{i+3}_x$ を次に示す曲線定義式(21)、(22)に代入する。

【0068】

$$x^i(t) = P^i_x N^i_0(t) + P^{i+1}_x N^i_1(t) + P^{i+2}_x N^i_2(t) + P^{i+3}_x N^i_3(t) \quad \dots (21)$$

$$y^i(t) = P^i_y N^i_0(t) + P^{i+1}_y N^i_1(t) + P^{i+2}_y N^i_2(t) + P^{i+3}_y N^i_3(t) \quad \dots (22)$$

【0069】次にS45で式(21)、(22)を次式(23)、(24)に示すような多項式の形に整理して

$$x^i(t) = A^i_x t^3 + B^i_x t^2 + C^i_x t + D^i_x \quad \dots (23)$$

$$y^i(t) = A^i_y t^3 + B^i_y t^2 + C^i_y t + D^i_y \quad \dots (24)$$

【0071】その後、S46で y 軸平行フラグがONか否かのチェックを行ない、OFFの場合は次の処理を行なう。

【0072】 $x^i(t)$ は高々3次関数であるので、簡

$$dx^i(t)/dt = 3A^i_x t^2 + 2B^i_x t + C^i_x \quad \dots (25)$$

【0074】この1次導関数=0の式をS47で2次方程式の解の公式を用いて解き、その t の値を求める。

【0075】次のS48で、この t の値をS45で求めた多項式(23)に代入することで、 $x^i(t)$ の極大値、極小値を求める。更に、S49で制御点区間 i に対応するノットベクトル t_i, t_{i+1} の値をS45で求めた多項式(23)に代入することにより、制御点区間 i の両端に於ける値を求める。

多項式の係数 A^i, B^i, C^i, D^i を算出する。

【0070】

単に次式(25)に示す1次導関数を求めることができる。

【0073】

【0076】その後、S50で、S48、S49で求めた値を比較することにより、制御点区間 i に於ける x の最大値、最小値を求める。S50の処理が終了すると、S51で変数 i を1増加させ、次の制御点区間 i について同様の処理を行なう。

【0077】これらの処理を全制御点区間、即ち曲線全体で行なうことにより、図5のS35の判定結果がYESとなり、S36の処理が行なわれる。

【0078】S36ではy軸平行フラグがONかOFFかをチェックする。今の例ではy軸平行フラグはOFFになっているので、S37の処理が行なわれ、曲線全体でのxの最大値、最小値が求められる。その後、S38でxの最大値、最小値とペアになるyの値を算出し、次のS39でxの最小値とそのペアになるyの値を始点座標値、xの最大値とそのペアになるyの値を終点座標値とした直線データを演算装置1内のメモリに格納し、その後、処理を終了する。

【0079】また、y軸平行フラグがONの場合も同様に、図6のS52～S55で各制御点区間のyの最大値、最小値を求め、図5のS40～S42で直線データを演算装置1内のメモリに格納した後、処理を終了する。

【0080】

【発明の効果】以上説明したように、本発明は、制御点が直線上に順列に並んでいない場合、各制御点区間毎の曲線定義多項式とその1次導関数の式とに基づいて全制御点区間に於ける最大値、最小値を算出し、更に、全制御点区間の最大値、最小値に基づいて直線の始点、終点を特定するようにしたものである。制御点が直線上に順列に並んでいない場合であっても、直線の基本データである始点、終点の座標値を特定することが可能になる効果がある。

【図面の簡単な説明】

【図1】本発明の実施例のハードウェア構成を示す外観図である。

【図2】本発明の処理内容の要点を説明するための流れ

図である。

【図3】本発明の実施例の詳細な処理内容を示す流れ図である。

【図4】本発明の実施例の詳細な処理内容を示す流れ図である。

【図5】本発明の実施例の詳細な処理内容を示す流れ図である。

【図6】本発明の実施例の詳細な処理内容を示す流れ図である。

【図7】従来、考えられる方法の処理内容を示す流れ図である。

【図8】2次元点か否かの判定方法を示す流れ図である。

【図9】2次元直線が否かの判定方法を示す流れ図である。

【図10】Bースプライン曲線の表現形式を示す図である。

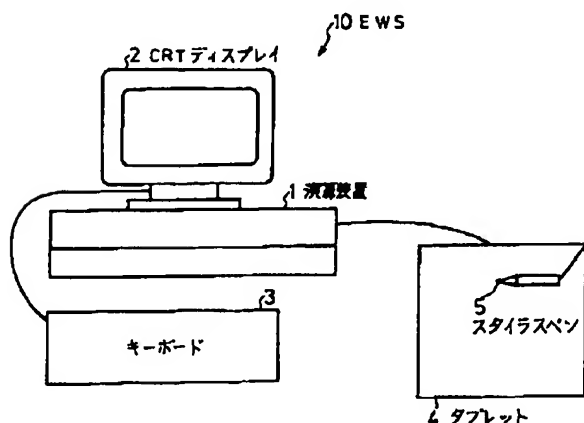
【図11】Bースプライン曲線が直線形状になる場合の例を示した図である。

【図12】Bースプライン曲線が直線形状になる場合の例を示した図である。

【符号の説明】

- 1…演算装置
- 2…CRTディスプレイ
- 3…キーボード
- 4…タブレット
- 5…スタイラスペン
- 10…エンジニアリングワークステーション（EWS）

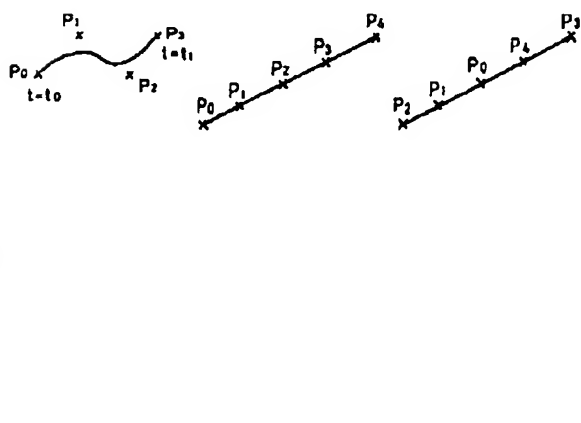
【図1】



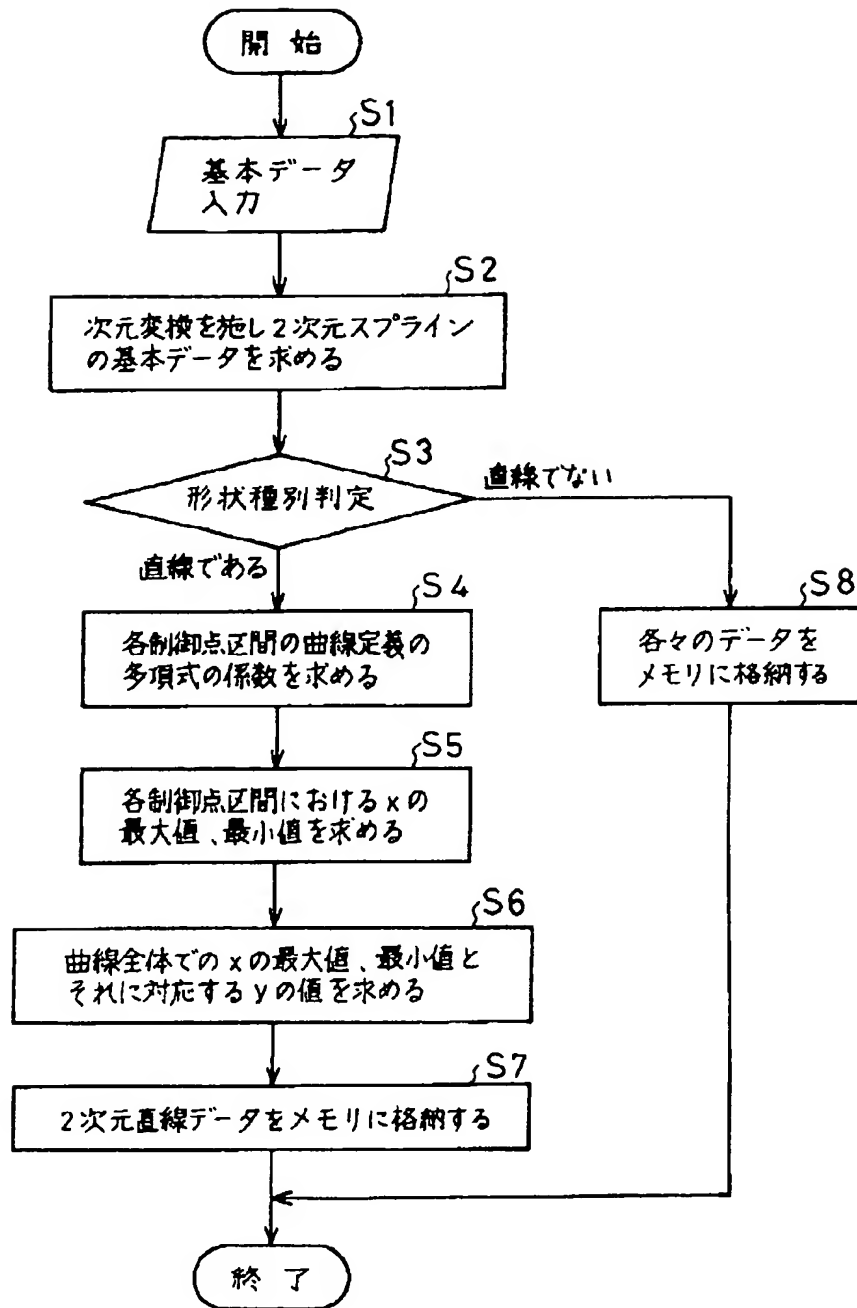
【図10】

【図11】

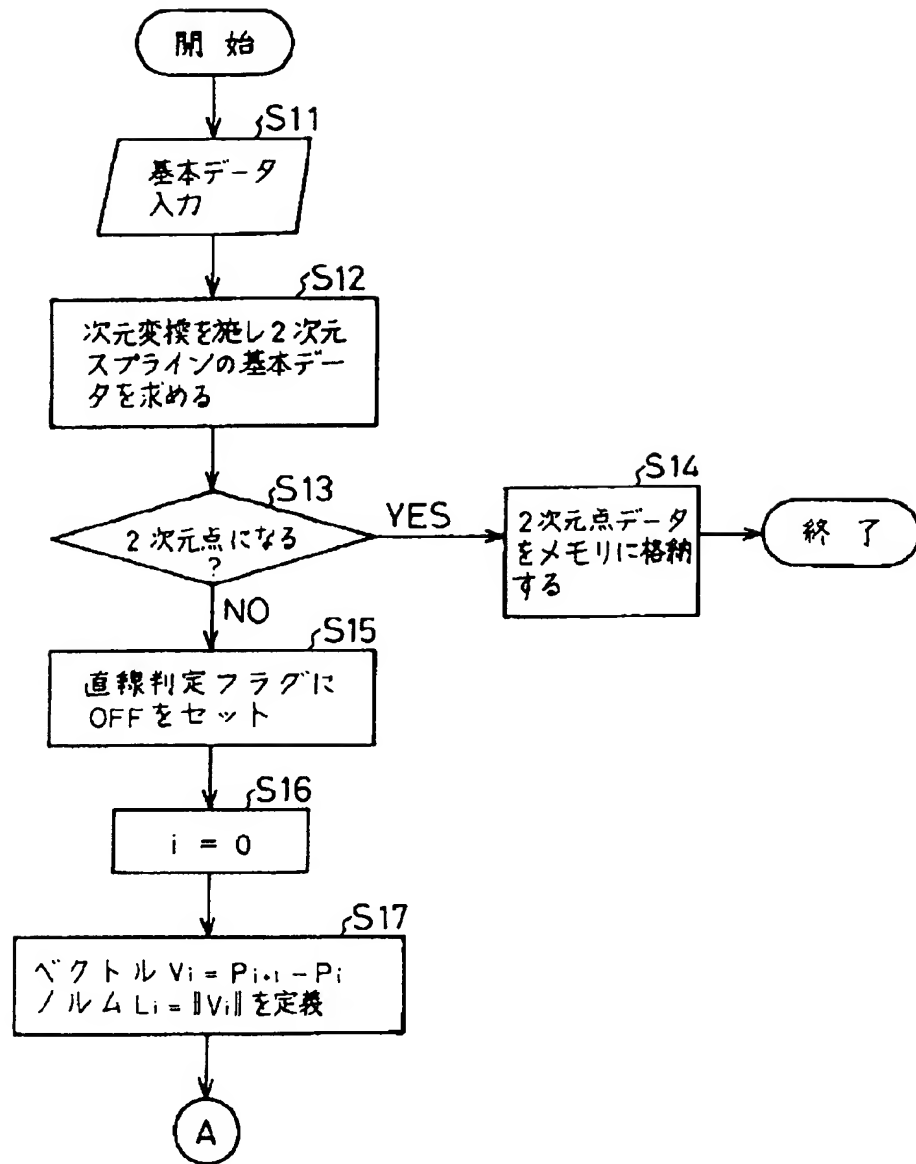
【図12】



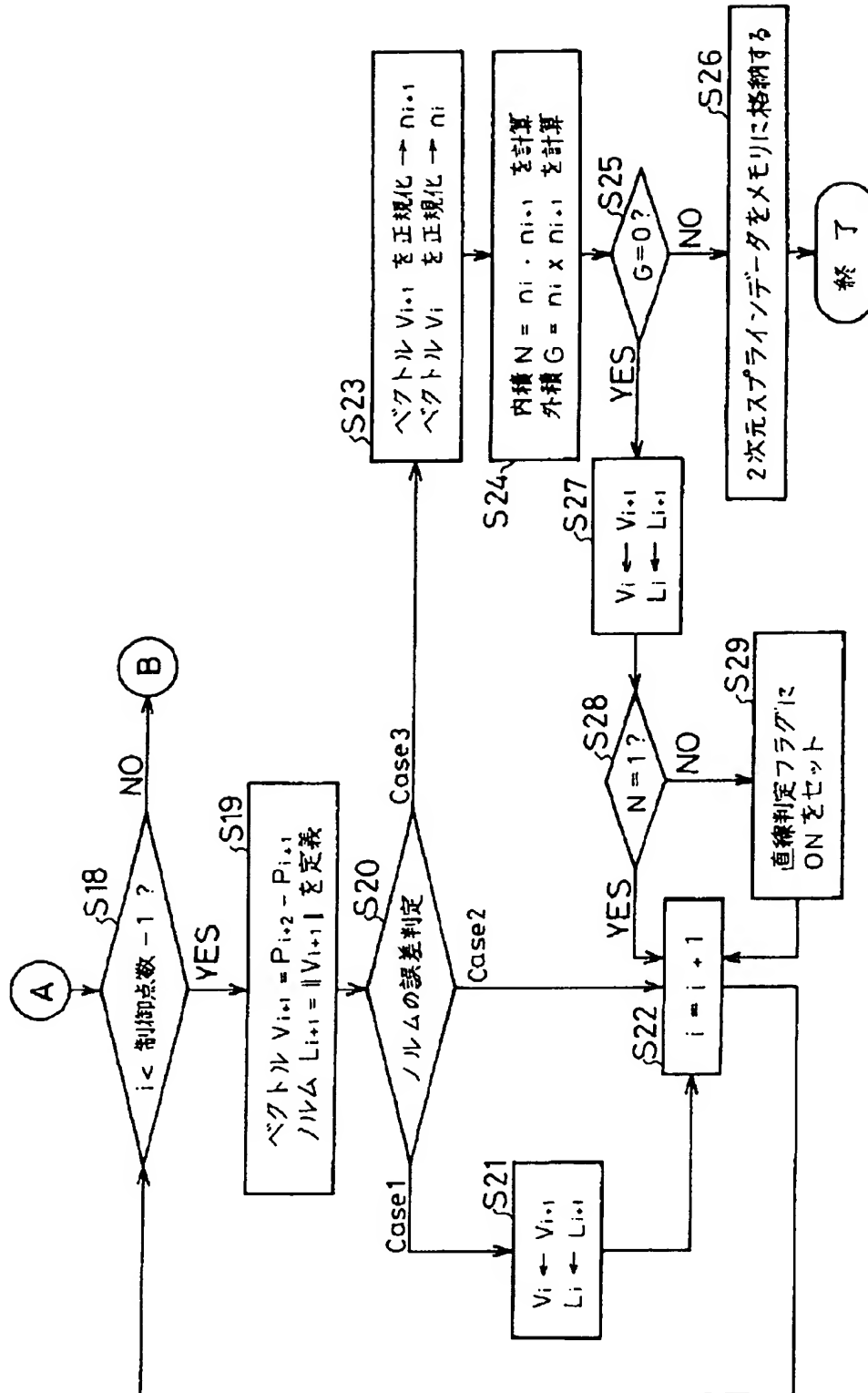
【図2】



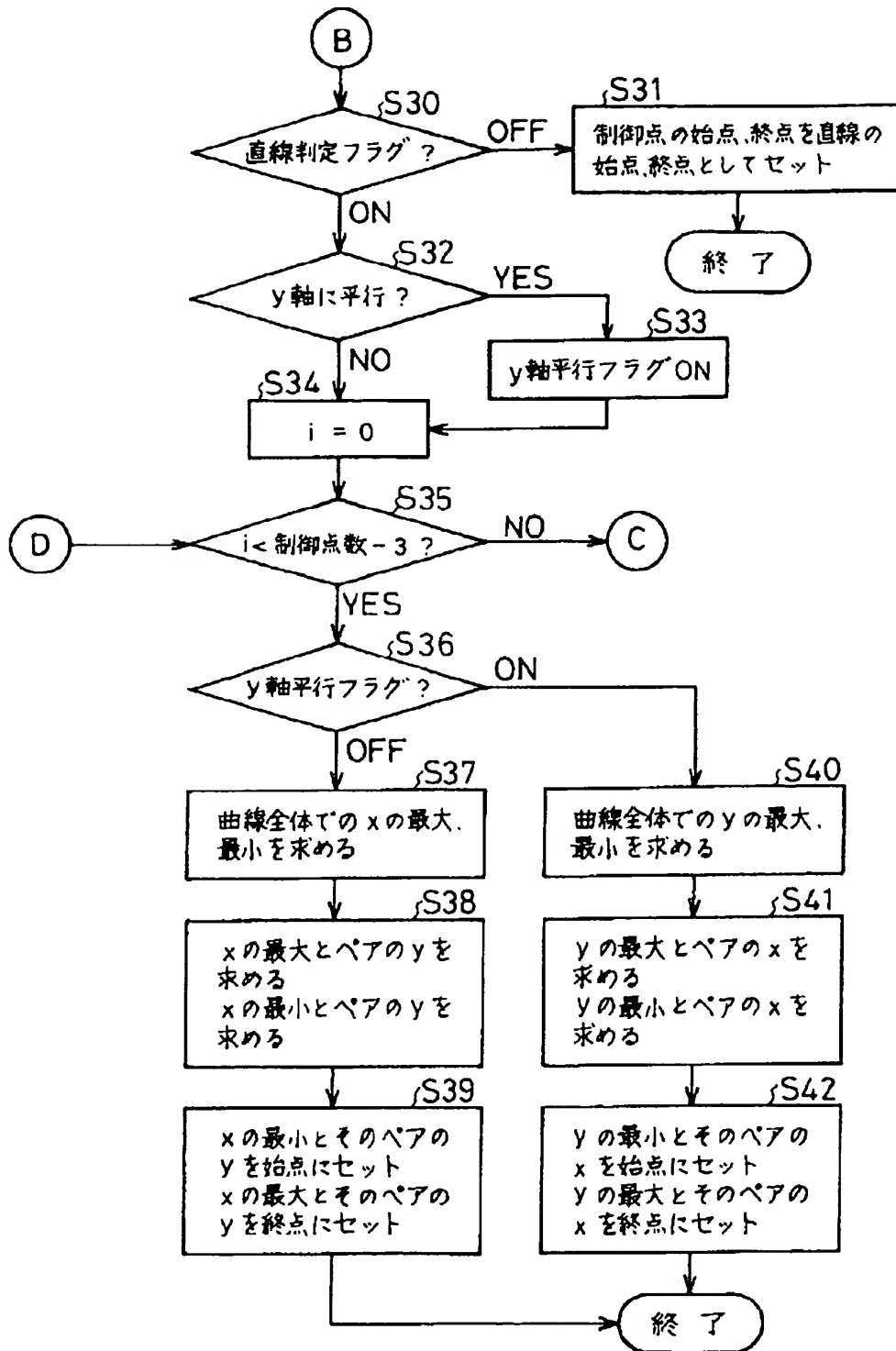
【図3】



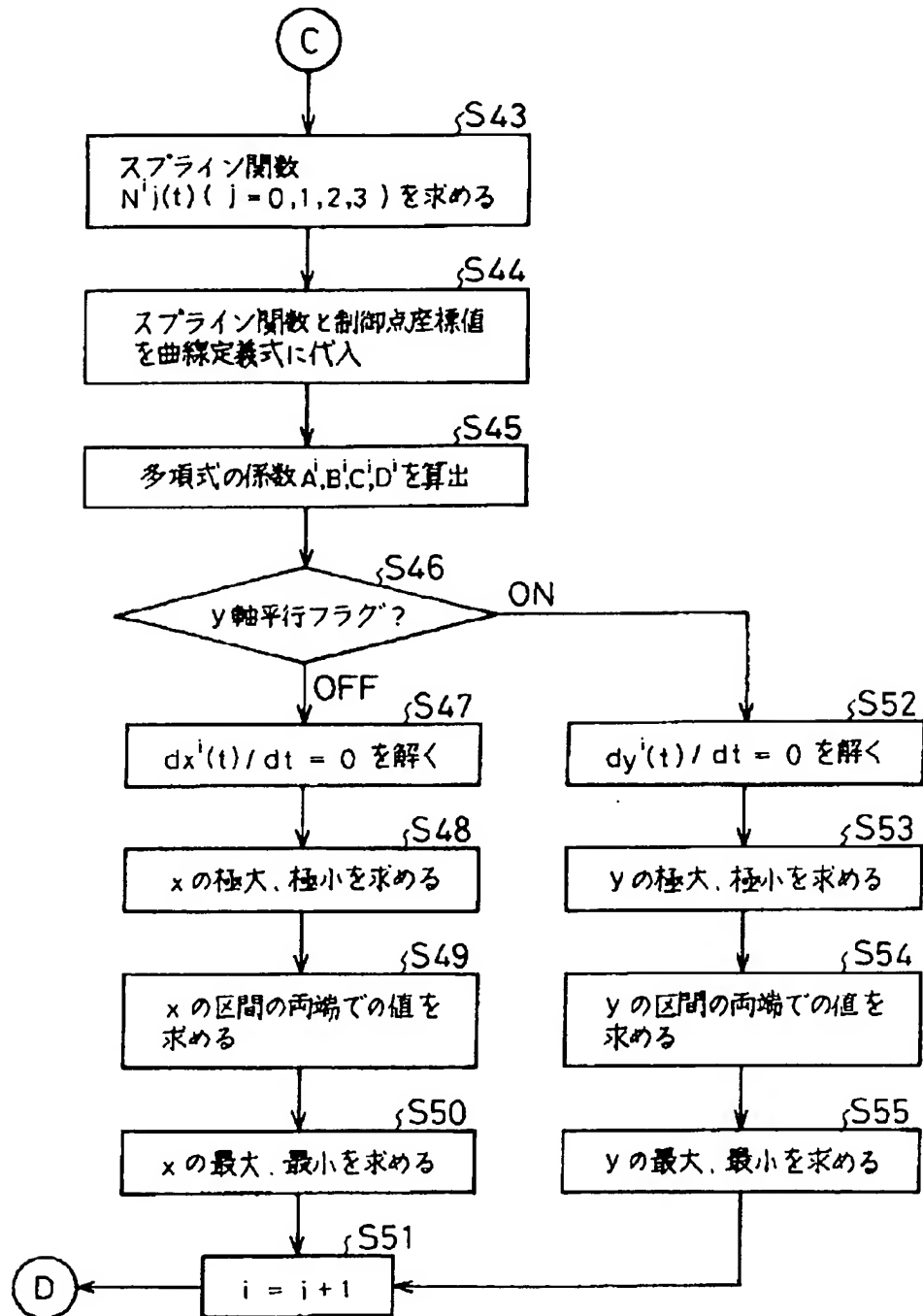
【図4】



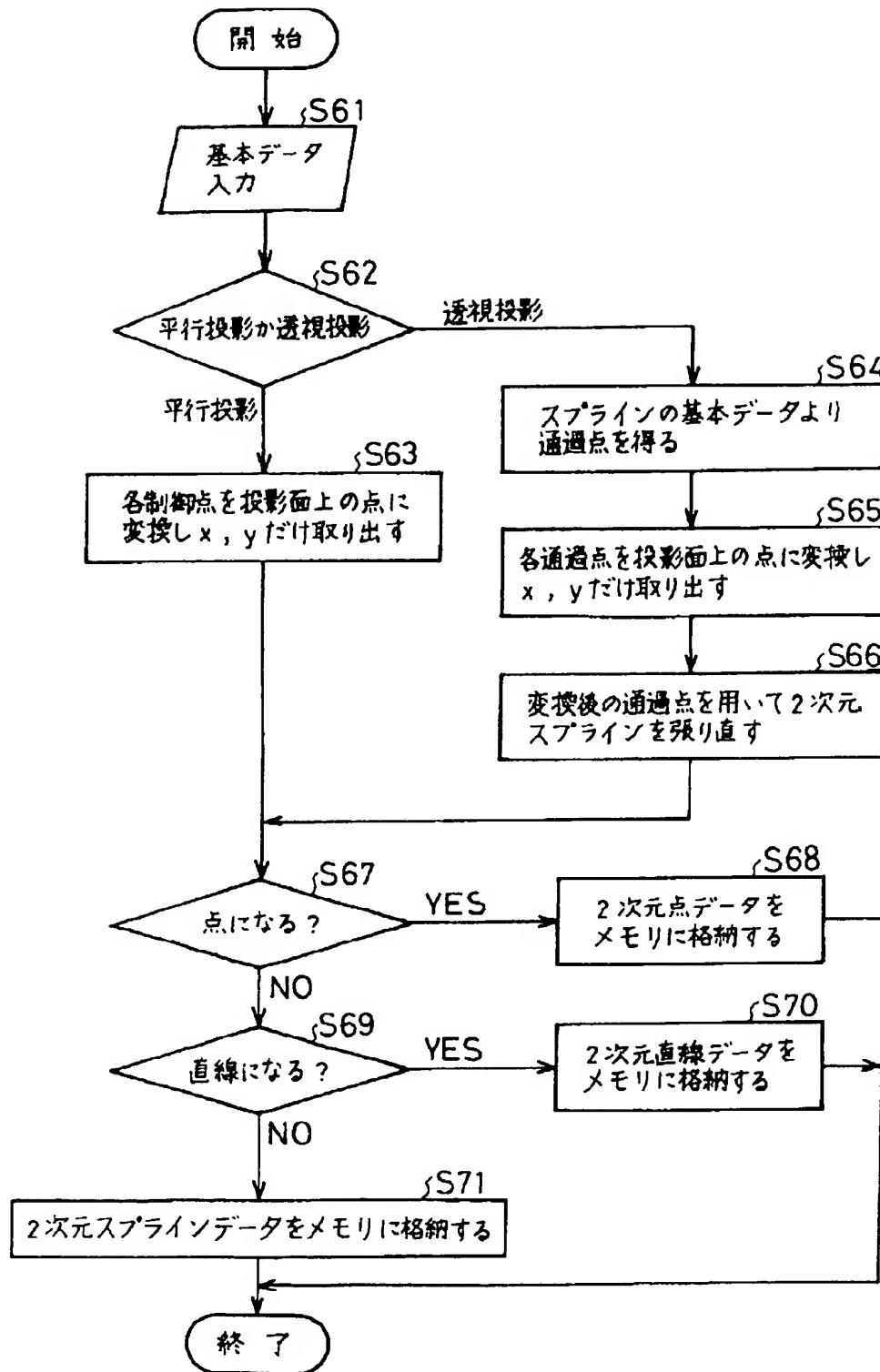
【図5】



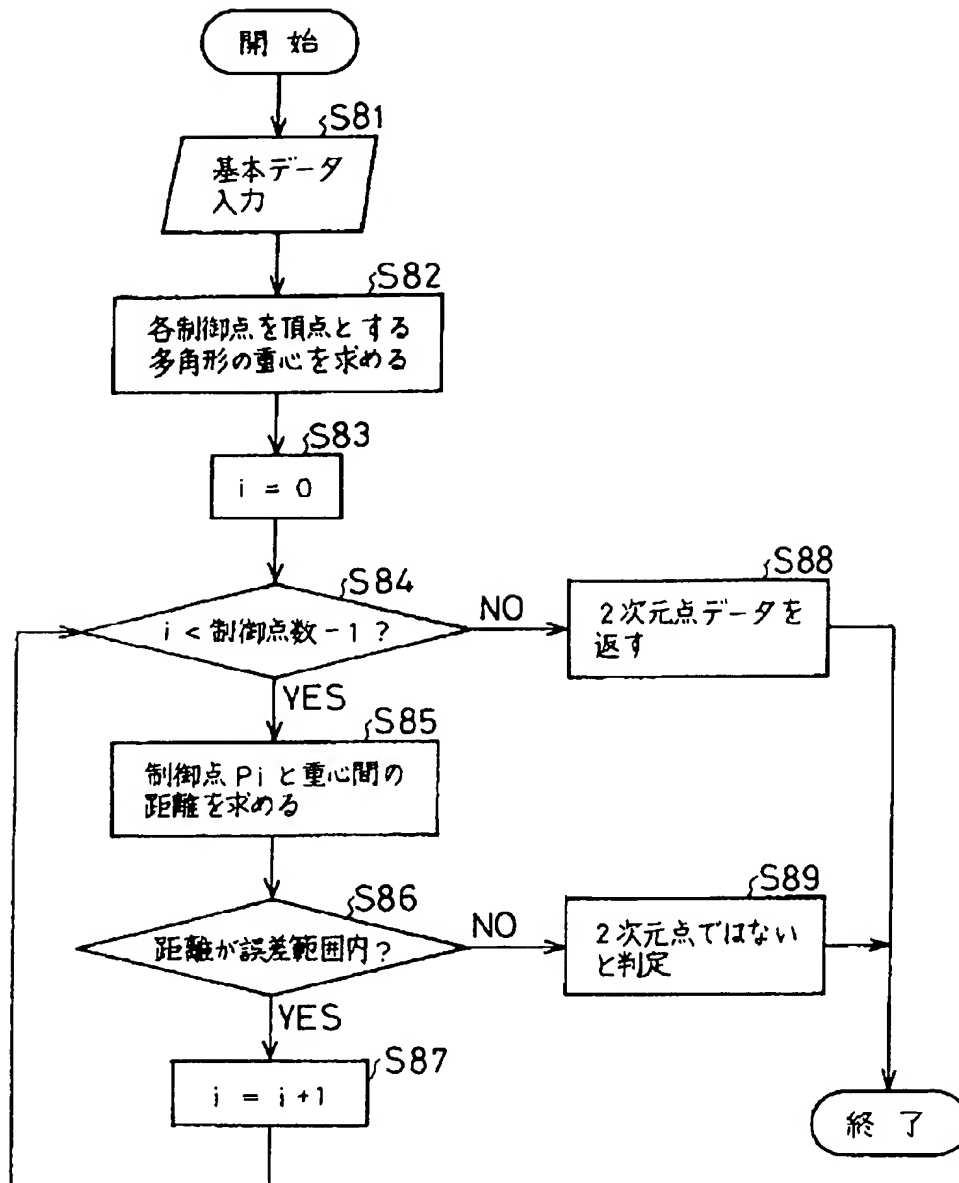
【図6】



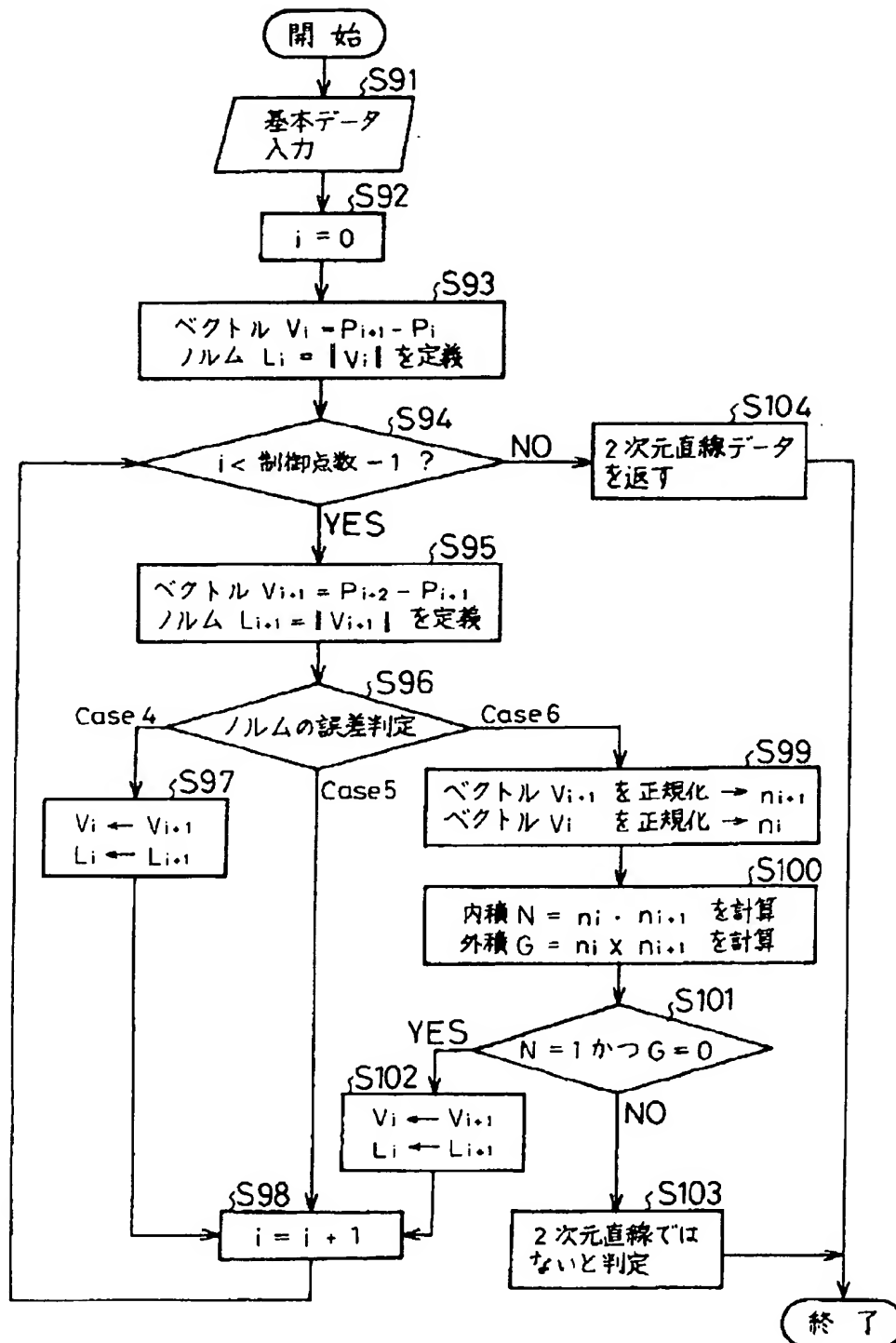
【図7】



【図8】



【図9】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.